LAB 7

| Question | Task | TIME ALLOCATION | REMARKs |
| --- | --- | --- | --- |
| 1 | More on constructors (copy and default), friend functions | 30 minutes |  |
| 2 | More on Friend functions | 90 minutes |  |

**Question 1**

1. Observe the given class and write the main() based on the instructions below:

#include<iostream>

using namespace std;

class Bags

{

string brand;

float height, length, width;

public:

void setdata()

{

cout<<"Enter your bag's brand name : ";

getline(cin, brand);

cout<<"Enter value length , width and height of your bag L, W, H ";

cin>>length>>width>>height;

}

void display()

{

cout<<"\nYour brand bag name is \*\*"<<brand<<"\*\* and the dimensions are: "

<<length<<"L "<<width<<"W "<<height<<"H "<<endl;

}

Bags (const Bags &Bi)

{

brand = Bi.brand;

length = Bi.length;

width = Bi.width;

height = Bi.height;

cout<<"\nDo you have the same bag??"<<endl;

}

Bags()

{

brand = "Adidas";

length = 35;

width = 20;

height = 45;

}

};

int main()

{ //need to developed by adding object K, L and M;

}

Complete the main():

1. Declare an object named **K**
   * Call member functions setdata() and display()
2. Declare another object named **L**
   * Call display()
3. **Compile the program and observe the output.**
4. Declare another object named **M** and initialize it with L.
   * Call member display()
5. **Compile the program and observe the output.**
6. Based on the solution at (a), create a friend function called ***check(….).***
   1. Parameters : object **a**, object **b**, object **c** of the class Bags.
   2. The function will determine if the 3 Bags objects share the same Height dimension and display the result.

In main() , create an array of 3 object elements. Using a for loop, invoke ***setdata(…)*** for each element. Lastly, pass the 3 object elements to the friend function call.

|  |
| --- |
| **Sample Output Screen 1 b) #1** |
| Enter your bag's brand name : adidas  Enter value length , width and height of your bag L, W, H 10 20 **30**  Enter your bag's brand name : puma  Enter value length , width and height of your bag L, W, H 15 25 **30**  Enter your bag's brand name : nike  Enter value length , width and height of your bag L, W, H 20 25 **30**  Do you have the same bag??  Do you have the same bag??  Do you have the same bag??  **Common height for all 3 bags** |

|  |
| --- |
| **Sample Output Screen 1 b) #2** |
| Enter your bag's brand name : adidas  Enter value length , width and height of your bag L, W, H 12 12 15  Enter your bag's brand name : nike  Enter value length , width and height of your bag L, W, H 20 30 20  Enter your bag's brand name : puma  Enter value length , width and height of your bag L, W, H 15 40 40  Do you have the same bag??  Do you have the same bag??  Do you have the same bag??  --not all bags have the same heights-- |

**Question 2**

1. Given the declarations for a class ***ICE\_CREAM*** that contains the following features:

#include<iostream>

#include<iomanip>

using namespace std;

class ICE\_CREAM

{

private:

string flavour;

int number;

float price;

public:

void menu();

void setflavour();

void setHowMany();

friend void display\_receipt(ICE\_CREAM);

ICE\_CREAM();

};

Write the definitions of the member functions **outside of the class** based on the desriptions given below:

1. **menu()**

Display the menu (refer to sample output screen)

1. **setFlavour()**

Get user ‘s choice and to set the *flavour* and *price* based on user’s choice. ( You have to use switch statement for this.)

1. **setHowMany()**

Get user input for *numbe*r.

1. **ICE\_CREAM( )**

Display “Barney’s House of Ice “.

1. Define friend function named **display\_receipt (…)** that accepts an object (from ICE\_CREAM class). In this function, display the payment details. (refer to sample ouput screen)
2. In main(), do the following:
3. Create an object of class ICE\_CREAM called **IC**
4. Call the required member functions using **IC**
5. Pass **IC** when making function call to **display\_receipt(…)**

|  |
| --- |
| **Sample Output Screen** |
| BARNEY'S HOUSE OF ICE  ===========================================  === CHOOSE FLAVOUR ===  ===========================================  [1] === Strawberry Flavour RM 3.50  [2] === Chocolate Flavour RM 2.50  [3] === Vanilla Flavour RM 1.50  [4] === Durian Flavour RM 0.50  Choice of flavour : ***2***  How many : ***6***  ===========================================  === PAYMENT ===  ===========================================  Flavour : Chocolate  Total Price : RM 15.00 |

**Question 3**

Based on the incomplete program given below:

|  |
| --- |
| #include<iostream>  using namespace std;  class NumberGame  { int array[5];    public:  //-----------3(a)-----------    friend void search(NumberGame , int\*);  };  //-----------3(b)-----------    int main()  { NumberGame G ;  int num;  cout<<"Enter a number :";  cin>>num;    search(G, &num);  return **0;**  **}** |

|  |
| --- |
| **Sample Output Screen Q3 #1** |
| Enter a number :***55***  55 is NOT found ! |

|  |
| --- |
| **Sample Output Screen Q3 #2** |
| Enter a number :***33***  33 is found ! |

1. Define default constructor to set the array with the following values.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 | 20 | 33 | 38 | 100 |

1. Define *friend* function ***search(…)*** to find a number in the array of the object of class *NumberGame*.
   * The function will receive two parameters (refer to the prototype given in the class above). The second parameter is a pointer which will receive the number from the function call by reference. This number will be used in the *do-while* loop.
   * Write a *do-while* loop to loop through every element of the *array* to search for the number.
   * Once the number is found, stop the search.
   * Using *if-else* statements, display the appropriate message for both situations (whether the number has been found or not).

**[Note: refer to sample output screen above]**